TOWARD USING INTELLIGENT AGENTS TO DETECT, ASSESS, AND COUNTER CYBERATTACKS IN A NETWORK-CENTRIC ENVIRONMENT

Martin R. Stytz, Ph.D.
Institute for Defense Analyses
Washington, DC
(703) 338-2997, (407) 497-4407
mstytz@ida.org , mstytz@att.net

Dale E. Lichtblau, Ph.D.
Institute for Defense Analyses
Washington, DC
(703) 845-6683
del@ida.org

Sheila B. Banks, Ph.D.
Calculated Insight
Orlando, Fl 32828
(407) 353-0566
sbanks@calculated-insight.com

Abstract

The network-centric warfare philosophy is becoming more firmly entrenched in US military doctrine and operations. As a result, the state and trustworthiness of the network and its computational resources are becoming even more important for commanders, particularly as the network itself is becoming an ever more lucrative target for cyber attack. In cyberspace, however, given human limitations and the fact that intelligent agents (computer viruses, worms, etc.) execute most cyber attacks, we argue that the netcentric environment will require computerized agents to detect, assess, and respond to cyber attacks. A significant portion of day-to-day network operations will have to be allocated to intelligent agents (or computer-generated forces (CGFs)). These CGFs will have to determine the types of attacks that are underway, the targets of the attacks, the appropriate responses to the attacks, the prioritization of the responses, the erection of defenses against secondary attacks, the response to the primary attack(s), and for the overall management of the response.

1. INTRODUCTION*

As the network-centric warfare philosophy becomes more firmly entrenched in US military doctrine and operations, the state and trustworthiness of the network and its computational resources will become important information for commanders, particularly since the network itself will become an ever more lucrative target for cyber attack. In cyberspace, however, both the pace and breadth of attacks humans require assistance in order to achieve timely attack analysis and response. The pace of events in cyberspace and attendant workload limits effective human involvement in determining the state of network resources and of the facilities attached to the network. Given human limitations and the fact that intelligent agents (computer viruses, worms, etc.) execute most cyber attacks, we conclude that the netcentric environment will require semi-autonomous agents to detect, assess, and respond to cyber attacks. A significant portion of day-to-day network operations will have to be allocated to intelligent agents (or computer-generated forces (CGFs)). These CGFs will have to determine the types of attacks that are underway, the targets of the attacks, the appropriate responses to the attacks, the prioritization of the responses, the erection of defenses against secondary attacks, the response to the primary attack(s), and for the overall management of the response.

The threat posed by the incentivized cyber attacker of the future is increased because the transition to network centric warfare promises to increase the effectiveness of future military operations, which makes the network and its software higher value targets. The promise of increased military effectiveness arises from the capability for network centric warfare to increase the effective combat power of military organizations. The increase occurs as a result of the provision of timely and relevant information organized and presented to facilitate situation awareness, decision-making, and response to enemy activity, friendly activity, and other circumstances. Network centric warfare can substantially reduce the fog and friction of war and thereby reduce the most serious impediments to optimal, effective action in the battlespace. As a result of the empowerment that results from improved, efficient information flows, commanders at all levels can effectively and efficiently employ and coordinate their resources and actions to achieve objectives and capitalize upon transient

* [1] The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of Defense or the US Government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
JUN 2005

2. REPORT TYPE

3. DATES COVERED
00-00-2005 to 00-00-2005

4. TITLE AND SUBTITLE
Toward Using Intelligent Agents to Detect, Assess, and Counter Cyberattacks in a Network-Centric Environment

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses,4850 Mark Center Drive,Alexandria,VA,22311

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	**unclassified**	**unclassified**		**21**	

opportunities in the battlespace to further increase their effectiveness and combat power. However, a central, but generally unspoken, condition for successful network centric warfare is that the information received is actionable; i.e., that the information is both timely and correct. However, the increasing sophistication of computer and network attack technologies and tools coupled with the increasing technical sophistication of potential adversaries calls this implicit central tenet into question and makes the question of how to secure the network and software against the threat of attack and subversion all the more urgent and important. Hence, our conclusion that the threat posed by cyber attack will increase as the transition to network centric warfare proceeds.

Before CGFs can be used effectively to counter cyber attacks, four significant technological developments must be achieved. These developments include: 1) development of an appropriate individual CGF architecture, 2) development of a distributed CGF system, 3) acquisition of the knowledge needed by the CGFs to perform their assigned activities, and 4) the development of cyber sensors that can acquire data about the state of the cybe rbattlespace so that the cyber conflict and cyberwarfare resources can be managed. So, even though knowledge about cyber attacks is growing and becoming better organized, the amount of information is so vast and uncorrelated that there is a need for the automated data acquisition, attack categorization, defensive response, and analysis that only CGFs can provide. In light of the magnitude of these four challenges, this paper addresses the challenge that we feel is critical: the acquisition and distribution of the data needed to manage the cyber conflict and allocate cyber resources. To effectively manage the cyber conflict and allocate cyber resources, the research and development challenge is to build a set of hybrid cyber sensor CGFs to be distributed throughout the network so that they can gather data about an attack as it is underway. The data gathered by the cyber sensors must enable cyber managers to categorize the attack, determine the potential severity of an attack, and provide a portal into the cyberbattle that permits human monitoring of the event and the response. In this paper we describe the CGFs for data acquisition and how they need to be assembled and deployed to address these tasks.

There are a number of requirements that we have identified that must be addressed in order for a cyber sensor CGF to be effective and to be able to transmit the data they accumulate to cyber command and control center(s). In addition to gathering data from individual locales where cyber activity is occurring, data such as the type of attack, attack payload contents, apparent attack strategy, and apparent attack origination point must be gathered. The task of the cyber sensor CGFs is further complicated by the fact that they must gather data about an attack whose components are separated in space and time. While some of the responsibility for determining stealthy cyber attacks must reside with the cyber command and control center, the cyber sensor CGFs must have enough intelligence and analytical capability to determine whether the events that they are witnessing are part of a larger stealthy attack. Furthermore, the cyber sensor CGFs must operate within a network using technologies that assure that their data reaches the command and control center rapidly and securely. As a result, some network bandwidth must be dedicated to cyber sensor data reporting and CGF authentication. Moreover, because a variety of cyber sensors is required, they must be able to migrate securely throughout the network (or at least their knowledge bases must migrate) in order to gather data and to aid in attack response. The cyber sensors must also be autonomous, or nearly so; therefore, they must operate using a set of action policies that guide them in their activities and in the type of data that they gather and report. Finally, the cyber sensors must be able to exchange data among themselves in a secure manner and in a manner that conserves network bandwidth and yet insures that all of the cyber sensors receive the data that they require from the other cyber sensors in the network, even during a cyber attack. Our work is intended to further refine these requirements and to outline solutions to these CGF cyber sensor challenges.

The remainder of this paper is organized as follows. Section Two contains a brief discussion of previous research related to the cyber sensor research problem. Section Three presents a more detailed discussion of the requirements for the CGF cyber sensors. Section Four presents a discussion of the manner in which distributed simulation technologies can be employed to refine cyber sensor CGF requirements and to validate the effectiveness of solutions. Section Five contains a brief summary of our findings and suggestions for further research in this important field.

2. PREVIOUS RESEARCH

Before discussing the solution we propose to the cyber security threat, we will briefly examine the different forms of cyber attack to illustrate the

ATTACK VECTOR	SPECIFIC STRATEGY	BROAD STRATEGY
Block Access to Libraries	Attack via environment.	Attack Through Runtime Environment
Redirect Access to Libraries	Attack via environment.	
Manipulate application registry values	Attack via environment.	
Force the application to use corrupt files or databases	Attack via environment.	
Manipulate and replace files that the application creates, reads, writes, or executes	Attack via environment.	
Force the application to operate in low memory, disk-space, and network-availability conditions	Attack via environment.	
Overflow input buffers	Attack through the user interface or other input vector.	Attack Through Source Code
Attack through application switches and options	Attack through the user interface or other input vector.	
Use escape characters, different character sets, and commands to get malformed input	Attack through the user interface or other input vector.	
Try common default and test names and passwords	Attack through design flaws.	
Look for and test unprotected application APIs	Attack through design flaws.	
Force the application to reset its values	Attack through default values.	
Get between time of check of a value and time of use of a value	Interposition attack.	
Create loop conditions in an application that reads script, code or other user supplied macros or logic	Attack through design flaws.	
Look for and use alternative execution routes through an application to accomplish its task(s)	Attack through design flaws.	
Connect to all ports	Attack through design flaws.	Attack Through Input Vectors
Fake the data source	Attack through design flaws.	
Create fake files with the same name as protected files	Attack through privilege.	
Force all error messages	Attack through privilege.	
Look for temporary files for an application and examine their contents for sensitive or exploitable information	Attack through files.	
Force invalid outputs to be generated	Attack through files.	
Attack through shared data	Attack through files.	

Table 1: Cyber Attacks and Their Strategies

scope of the threat. After reviewing the literature in this area[1-14], there is clearly no commonly accepted classification of attacks or the underlying strategies that are used to execute the attacks. So, in order to assist in understanding the scope of the threat and the techniques used to accomplish an attack, we developed our own classification of the types of cyber attacks and the strategies that they use. To insure that we captured all of the types of attacks and strategies, we used a successive refinement approach to distinguish and classify attacks and strategies. The classification was validated by a regular and thorough re-review of the literature to insure that the attacks and strategies that we identified captured all of the attacks and approaches taken to accomplish a cyber exploit.

As a by-product of our analysis of the cyber security attack literature, we identified three basic attack strategies, and we will open our discussion of the scope of the threat by presenting these strategies. The strategies are the following: 1) to inject faults via the application's runtime environment, 2) to inject faults through the application's source code, or 3) to inject errors via any of the application's input vectors in order to induce a fault or an application failure. The literature indicates that these strategies can be further refined and specialized. These basic strategies illustrate that the scope and basic approaches for cyber attack have not changed over the years and are relatively straightforward. While the strategies have not changed, there has been an increase in the sophistication and expertise employed to execute the strategy when performing an attack. These strategies can be executed using a variety of techniques and tactics, but some of the techniques required to execute them are quite complex. This set of strategies is useful for illuminating the scope of the cyber threat, as there are many types of cyber attack. In Table 1, we summarize the different types of attacks and provide a brief description of the strategy underlying each attack (exploit). There are a number of strategies identified in the table but they are all variants of the three strategies identified above. Based on this work and a survey of the literature, we were able to develop objectives and requirements for the CGF cyber defense system. We discuss these objectives and requirements next.

3. REQUIREMENTS

To lay the foundation for the discussion in this section, we present the chief objectives for cyber security. The objectives are the following: 1) preserve the integrity/functionality of the network and system; 2) control the use of the system; 3) prevent extraction of software subsets; 4) protect system data; 5) protect network access and prevent unauthorized access; 6) insure correct and accurate execution (unchanged processes that might still produce correct answers or incorrect answers); and 7) insure that computations are correct and accurate.

Consider that the basic strategic motivation for defense in depth is that no single defense should be relied upon to protect important items and instead multiple defenses should be employed. As a result, an attacker cannot defeat one defense and thereby gain access to the items being protected. Instead, all defenses must be defeated and, when properly arrayed, the attacker cannot gain insight into one defense while attacking another and, just as importantly, there is a degree of mutual support but not interdependence between defensive defenses. Therefore, mutual support combined with independence should be the objectives and guiding lights for achieving an effective defense in depth in the cyber world. As regards the cyber world, our problem is to interweave all of the defenses into one layer that would consist of mutually reinforcing but independent cyber defensive measures designed to keep a malicious event from occurring. Given the difficulty that the human mind has in maintaining an accurate mental conception of an environment or set of circumstances when seven or more items are in play simultaneously, the more defensive challenges that must be mastered simultaneously by an attacker, the stronger the defense should be and the more difficult it is to compromise.

We envisage an abundance of mobile, autonomous, and intelligent software agents whose function is to "roam" the network (from host to network node to network node…to host), pausing at each node, as necessary, to assess the node for vulnerabilities, to address vulnerabilities, to sense network activity for attack signatures, and, if warranted, take some counter-cyber attack action. We envisage a loosely-coupled coalition of cyber sensor CGFs whose sole purpose is to "protect and defend" the Global Information Grid (GIG). Their protective function is to "establish and maintain" a (logical) defensive perimeter. Their defensive function is to detect, analyze, and response as appropriate to "cyber attacks," whether intentional or inadvertent. On the protective side, they will be responsible for ensuring that access control mechanisms, firewalls, intrusion detection systems, and anti-virus and anti-pest software are appropriately deployed and enabled. They will also scan for known vulnerabilities and assess risk. On the defensive side, these CGFs must identify possible attacks, singly or collectively analyze the event in terms of probable source, degree of potential adverse effects—and hence, risk—develop courses of action, and, finally, respond as most appropriate.

As is well known, network operations will require the pervasive use of autonomous software agents to counter cyber attacks on the Global Information Grid (GIG). Before these agents—computer-generated forces—can be effectively deployed, four significant developments are needed:

- A CGF (agent) architecture
- An architecture for a system of these (distributed) CGFs
- A knowledge management system—sensor data collation, analysis, and distribution—to enable the operation of the system of distributed CGFs (see part III below)
- An assortment of cyber sensors that can acquire and report network cyber attacks

The architecture requirements of the cyber sensor CGFs fall into four categories: mobility, sensing, analysis, and communication. The intelligent software agents we are proposing must be mobile in the sense of transmitting themselves across the network, from one network node (host or network device) to another. The reason for this mobility requirement is two-fold. First, it would not be practical, even if feasible, to deploy a status CGF at every network node on which a cyber sensor ought to reside. Second, we want a dynamic defensive posture with cyber forces being dispatched (or autonomously dispatching themselves) to areas of the network that have come under attack or become more vulnerable as the network undergoes constant reconfiguration.

Unfortunately, in spite of numerous efforts undertaken to develop processes and technologies to enhance cyber security, we are far from being able to reliably achieve the goals listed above. Furthermore, no silver bullet solution to the problem of cyber security has been found and none appear to be on the horizon. As a result, we would argue that researchers and developers should re-examine the application of the idea of defense in depth and determine how it can provide better security for an application than a single defense or defensive layer. This straightforward idea appeals to our common sense and is also supported by hundreds of years of security experience in a variety of situations; ranging from national defense to military fort construction. However, most, if not all, of these systems were and are serial (or sequential) in nature. In other words, breaking one system opened the way to the next system, but until the first system was breached the second layer did not come into play. In the physical world, this approach to defense in depth is logical and effective. The nature of the physical world makes a sequential defensive system effective since the attacker cannot begin to devise an attack upon the inner defenses until the outer defenses are breached. However, the cyber world is different and reconsideration of how defense in depth should be applied is warranted.

4. USE OF DISTRIBUTED SIMULATION ENVIRONMENT

In the longer term, the research focus must shift to the development of techniques and secure integrated development environments that protect software from the moment of its inception throughout its lifecycle and that automatically insert protection techniques in response to defined requirements and expected computational load. The longer-term research focus should anticipate that computational capability and networking capability will increase at their historical rates and exploit these computational breakthroughs to increase the security of application software. Therefore, the development of autonomous defensive capabilities for software and the capability for software distributed across a wide area network to coordinate their defense appears to be an important and attainable research objective. Also, given the drive to compose software applications from objects and components, research to determine how to secure applications whose parts come from a variety of sources, sources that must be assumed to be untrustworthy, is needed. Another long-term research objective that is apparent is the need for a capability for software to autonomously alter the operation of its protection mechanisms or even insert/activate new mechanisms in response to attempts to compromise the software being protected. Finally, refinement of software architectural and design methodologies to incorporate considerations and techniques for the development of secure software as well as evaluations of the utility of design methodologies to support the incorporation of software protection techniques are needed. All of this activity needs to be under the control of the cyber sensor CGFs that monitor the network activity, and under the control of the cyber defense network operations center.

Given these near and long term research focuses for software protection, we can identify four main application security research thrust areas. These four security areas are the following: 1) algorithm development, 2) software development environments, 3) benchmark and metric development, and 4) integration. The algorithms area addresses the need to develop improved algorithms and techniques for application security, which includes improved algorithms for watermarking, obfuscation, performance degradation, and other techniques. The software environment development focus area addresses the need for software development environments to enable development of protected software and that automatically create software pedigrees, automatically insert protection techniques into software under development, and otherwise protect software throughout its entire development process. The benchmarks and metrics area addresses the need for development of means to measure the strength and potency of different application security techniques, their impact on performance, their stealthiness, and other measures of the ability of an application security technique to protect software on a given computational platform. The fourth focus area, integration, addresses the need for research into the development of techniques for integration of multiple application security techniques into software to be protected in an efficient manner and also of the need for research into means for integration of application security techniques with network security and host operating system security in order to form an integrated software protection triad.

The development of techniques for detecting if the cyber space is under attack will require the placement of protection technologies within cyber space that can determine if an attack on the data is underway and also control the activation of the protection technologies and protection response. Given the expected cost of developing these capabilities, we believe that a more effective and efficient approach to providing cyber security may be to build a testbed that can be used to evaluate defensive options. The testbed can be used to develop technologies to determine if an attack is underway and how to manage the cyber space of the protection techniques and protection response. To control costs, the testbed should have a standard and secure interface so that any application can connect to the testbed and make use of its capabilities for evaluating defenses.

In the same vein, there is a need for application security test suites to be run in the testbed for assessing protection technologies. The test suites must include benchmarks, metrics, an evaluation system (or rating system), and security testing scenarios to allow the cyber defense community to evaluate the efficacy of different software protection techniques in cyber applications and environments. Metrics for resilience and protection are particularly important because they provide insight into the strengths and weaknesses of a given protection technique as well as the estimated time to defeat a technique. Closely tied to the need for standard application security test suites is a cost/benefit analysis of different software protection technologies within the cyber domain. Currently, there is no information available that relates the costs of software protection, which include implementation time, maintenance cost, and computational power consumed, to the benefits of a protection technique, which include the time required to defeat a particular protection technique. Also, standard test suites and metrics would allow for comparison across and among classes of techniques.

To be extensible, the testbed environment should provide a component-based framework enabling the opportunity to create, develop, test, and validate techniques and the models themselves within simulations of the environment where they will be fielded. The testbed environment should unify the research, development, and transition efforts of cyber space research and focus on the requirements. The testbed development environment will enable progress via a coherent vision for cyber protection and will not infringe or stifle innovative ideas within the basic research community. These basic research programs are a vital foundation to technology development; however, to utilize basic research for cyber space requires a common framework that provides processes, model development methodologies, and architectures not generally associated with basic research. For cyber space, the research knowledge gained by the basic research community must be focused into a useful work environment for mission rehearsal, training, analysis, and acquisition. By providing this focus, the testbed development environment provides coherent vision, transition opportunity, and real use coupled with extended development of cyber space research.

There are two major functions that the testbed environment should enable: (1) the integration of cyber defense techniques into the environment to allow for integration, testing, and analysis; and (2) the development of new techniques to enable the testing of new methodologies/theories for cyber defense and to allow the seamless integration of the model development process with the model execution and performance within a selected environment. After careful evaluation of the functionalities discussed and the available techniques, we determined that both functions were equally important for the testbed environment and that both would require the same framework to house these integration and building efforts. Therefore, the most general approach for the development environment architecture should be taken to thereby support extendable functionality and modifiability throughout the wide variety of model building and use conditions.

The architectural framework for the testbed must support incorporation of the following attributes: the development of consistent testing and domain metrics to allow for a seamless transition to the design, development, and testing environment within the environment. The testbed framework should also support the incorporation and direct use of existing cyber defense architectures and models to allow for a quick start in advancing model development efforts; and finally an intelligent interface capability that utilizes a cyber space ontology and common representation of knowledge to enable CGF development. The framework for the development environment must also provide software flexibility that includes model toolsets that encompass the inclusion of legacy software, knowledge acquisition and knowledge discovery integration, verification and validation processes, and testing tools. As a result of these considerations, we determined that the development environment should be component-based and framework-based.

5. SUMMARY AND FUTURE RESEARCH

The transition to network centric warfare places a premium on cyber security. The cyber security need has been poorly addressed to date. In cyber space, given human limitations and the fact that intelligent agents (computer viruses, worms, etc.) execute most cyber attacks, we argue that the netcentric environment will require cyber sensor agents to detect, assess, and respond to cyber attacks. A significant portion of day-to-day network operations will have to be assigned to intelligent agents or computer-generated forces. These CGFs will have to determine: the types of attacks that are underway, the targets of the attacks, the appropriate responses to the attacks, the prioritization of the responses, the erection of defenses against secondary attacks, the response to the primary attack(s), and for the overall management of the response.

There are a number of requirements that we have identified that must be addressed in order for a cyber sensor CGF to be effective and to be able to transmit the data they accumulate to cyber command and control center(s). In addition to gathering data from individual locales where cyber activity is occurring, data such as the type of attack, attack payload contents, apparent attack strategy, and apparent attack origination point must be gathered. The task of the cyber sensor CGFs is further complicated by the fact that they must gather data about an attack whose components are separated in space and time. While some of the responsibility for determining stealthy cyber attacks must reside with the cyber command and control center, the cyber sensor CGFs must have some intelligence and analytical capability. Furthermore, the cyber sensor CGFs must operate within a network using technologies that assure that their data reaches the command and control center rapidly and securely. Finally, the cyber sensors must be able to exchange data among themselves in a secure manner and in a manner that conserves network bandwidth and yet insures that all of the cyber sensors receive the data that they require from the other cyber sensors in the network, even during a cyber attack. Our work is intended to further refine these requirements and to outline solutions to these CGF cyber sensor challenges. Based on these considerations, we conclude that the development of a testbed for cyber defense CGF development is needed and we discussed the requirements for such a system.

The next steps for our CGF cyber research effort are to refine the requirements for the testbed and the distributed CGF and to assemble the prototype. We are currently evaluating potential software systems that will be protected by the CGF cyber defense system. We plan to report on the results of these efforts in future papers.

References

[1] Alexander, I. (2003) "Misuse Cases: Use Cases with Hostile Intent," *IEEE Software*, vol. 20, no. 1, January, pp. 58-66.

[2] Amoroso, E.G. (1994) *Fundamentals of Computer Security Technology*. Prentice Hall: Englewood Cliffs, NJ.

[3] Collberg, C.; Thomborson, C.; and Low, D. (1998) "Manufacturing Cheap, Resilient, and Stealthy Opaque Constructs," *Principles of Programming Languages 1998, POPL'98*, San Diego, CA, January.

[4] Denning, D.E. 1999) *Information Warfare and Security*, Addison-Wesley: Reading, MA.

[5] Garfinkel, S. and Spafford, G. (1991) *Practical Unix Security*. O'Reilly & Associates: Sebastopol, CA.

[6] Gollmann, D. (1999*) Computer Security*. Wiley: Mew York.

[7] Howard, M. and LeBlanc, D. (2002) *Writing Secure Code*. Microsoft Press: Redmond, Washington.

[8] Jalal, F. and Williams, P. (1999) *Digital Certificates*: *Applied Internet Security*. Addison-Wesley: Reading, MA.

[9] National Security Council. (1999) *Trust in Cyberspace*. National Academy Press: Washington, DC.

[10] Schneer, B. (1996) *Applied Cryptography*, John Wiley and Sons: New York.

[11] Stallings, W. (1999) *Cryptography and Network Security: Principles and Practice*. Prentice Hall: Upper Saddle River, NJ.

[12] Summers, R. (1997) *Secure Computing: Threats and Safeguards*. McGraw Hill: New York.

[13] Shrobe, H. (2002) "Computational Vulnerability Analysis for Information Survivability," *AI Magazine*, vol. 23, no., 4, Winter, pp. 81-91.

[14] Waltz, E. (1998) *Information Warfare: Principles and Operations*. Artech House: Norwood: MA.

[15] Howes, Norman R., Mezzino, Michael, Sarkesain, John, "On Cyber Warfare Command and Control," Proceedings

of the 2004 Command and Control Research and Technology Symposium, June 15-17, 2004, San Diego, CA.

TOWARD USING INTELLIGENT AGENTS TO DETECT, ASSESS, AND COUNTER CYBERATTACKS IN A NETWORK-CENTRIC ENVIRONMENT

Martin R. Stytz, Ph.D.
Institute for Defense Analyses
Washington, DC
mstytz@ida.org
,mstytz@att.net

Dale E. Lichtblau, Ph.D.
Institute for Defense Analyses
Washington, DC
del@ida.org

Sheila B. Banks, Ph.D.
Calculated Insight

Orlando, FL
sbanks@calculated-insight.com

Introduction

- "The battlefield is the computer"

- The bad guys have many motivations for attacking computational resources
  - Psychological, military, and financial
- Threat will increase
- So, our primary NCO resource is also a prime target

Introduction (cont.)

- Network Centric Warfare (NCW) increases effectiveness by information-based empowerment
- Increased power from information leads to increasing reliance on information
  - Unspoken tenet of NCW is that information is accurate
  - The growing threat brings this assumption into question because information will be attacked
  - Growing sophistication and effectiveness of cyberbattlespace offensive activity
  - Technical sophistication required to manage/conduct defense
- State and security of network will be critical to commanders
- Speed and complexity of cyberspace indicate that new defense approaches are needed

Cyber Battlespace Arena

- Events occur at high speed, much faster than human thought processes
- Rapid change in attack vectors
- Need for technical expertise for command and control
- Difficult to develop and maintain situation awareness
- Current lack of metrics to measure defense effectiveness
- Difficult to predict future activity in cyberbattlespace
  - No predictive battlespace awareness
- High degree of vulnerability to intended and serendipitious effects of cyberspace actions

CGFs

- In light of the types of attacks, what response should be made?
  - Preserve integrity/functionality of network
  - Control system use
  - Prevent extraction of software subsets (piracy)
  - Protect data
  - Protect network access
  - Insure correct and accurate software
  - Insure computations are correct and accurate
- Resultant CGF Capability Needs
  - Architecture
  - Distributed system (scale)
  - Knowledge acquisition
  - Cyber sensors
  - Most important task is knowledge acquisition for defense management

Framework for Analysis of Attack

- Goals, effort, vector
  - CGFs must be aware of all three
- Goals of attacks
  - Reverse engineering all or parts of a code
  - Allowing limited or unrestricted execution
  - Tampering with the code
- Type of effort needed for successful attack
  - Human effort (from expert to ordinary skills)
  - Generic tools (COTS, open source)
  - Specialized tools (what is possible by skilled adversaries?)
  - Number of allowed executions
  - Time and availability of code required for attack
- Vector for attack
  - Specific vulnerability exploited; means for delivering attack payload

Attack Identification Methodology

- **Identify each type of attack/exploit category**
  - Web and literature survey
  - Narrative description
- **Convert each narrative into UML threat case and sequence diagrams**
  - Threat case diagrams to document threats
- **Parallel development**
  - Tests, scenarios, and experiments to validate uncovered attacks
- **Testing and analysis of identified attacks and included major and minor threat cases**

Attack Classification

- No generally accepted classification
  - Developed classification based upon extensive research and correlation of literature
- Literature shows it is broad and growing
- Three basic attack strategies
  - Fault injection via environment
  - Fault injection through source
  - Fault injection via errors

Types of Attacks

1- Block Access to Libraries

2- Redirect Access to libraries

3- Manipulate application registry values

4- Force the application to use corrupt files or databases

5- Manipulate and replace files that the application creates, reads, writes, or executes

6- Force the application to operate in low memory, disk-space, and network-availability conditions

7- Overflow input buffers

8- Attack through application switches and options

9- Use escape characters, different character sets, and commands to get malformed input

10- Try common default and test names and passwords

11- Look for and test unprotected application APIs

12- Connect to all ports

13- Fake the data source

14- Create loop conditions in an application that reads script, code or other user supplied macros or logic

15- Look for and use alternative execution routes through an application to accomplish its task(s)

16- Force the application to reset its values

17- Get between time of check of a value and time of use of a value

18- Create fake files with the same name as protected files

19- Force all error messages

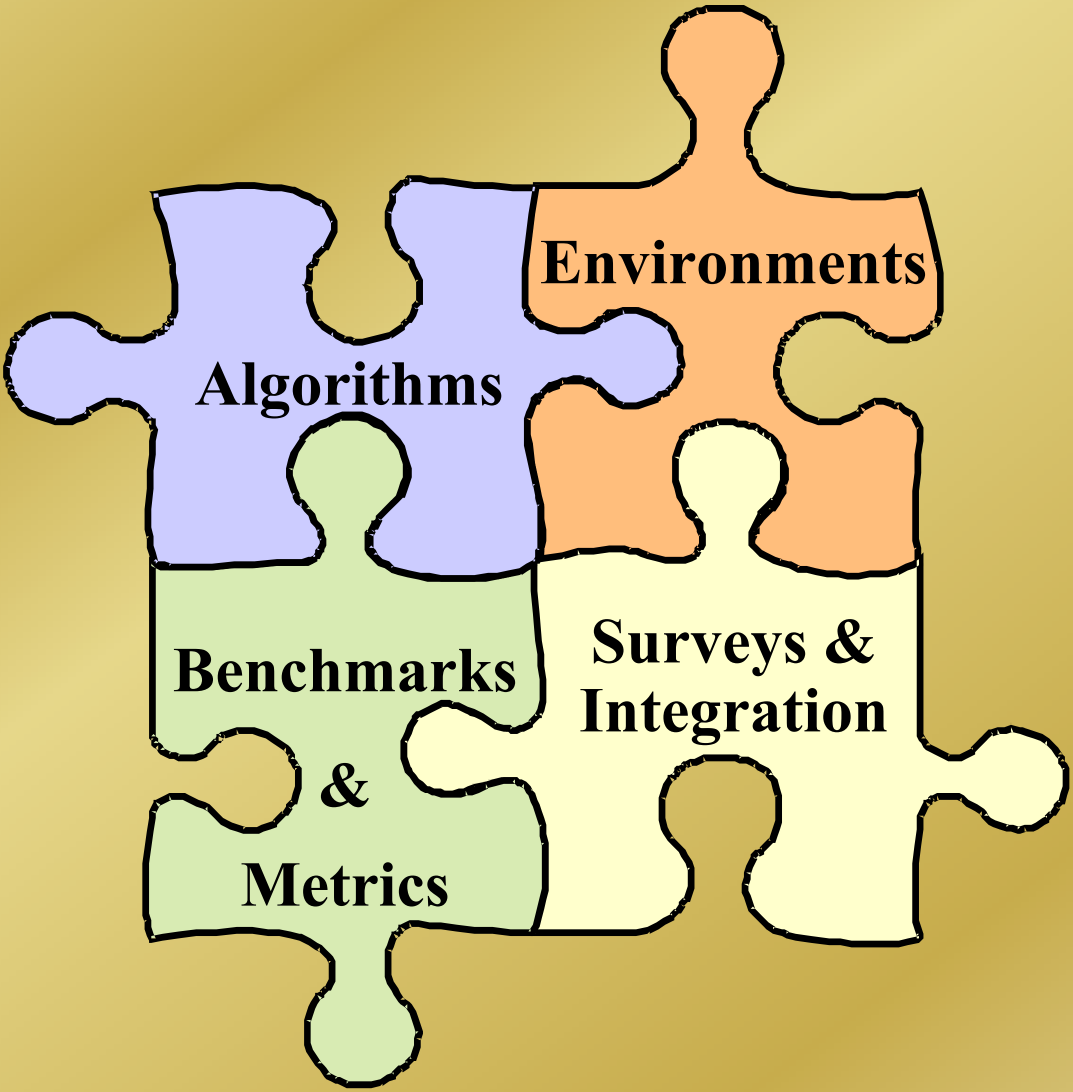
20- Look for temporary files for an application and examine their contents for sensitive or exploitable information

21- Force invalid outputs to be generated

22- Attack through shared data

- Block library access
- Overflow input buffers
- Connect to all ports
- Force error messages

Basic Research Requirements



CGF Cybersensor Requirements

- Data acquisition about local attack
- Identify type of attack, attack payload, strategy
- Attack origination
- Must be able to identify an attack and differentiate it from a system failure or fault
- Secure transmission of data from sensor to control sensor
- Secure migration
- Autonomic operation
- Exchange data among cybersensors securely
- Scan for vulnerabilities and assess risk

Addressing the Need

- Must develop techniques and environments to assemble the CGF cybersensorss
- Must test the CGFs as well
  - Real world too dangerous
  - Simulation environments provide protection for real-world and required complexity for CGF testing
- Develop application security test suites
- Build testbed for development and evaluation of technologies and CGFs
  - Secure development
  - Benchmarks, metrics, scenarios
  - Integrated cyber defensive techniques for testing and analysis
  - Techniques for testing of methodologies
- Need cost-benefit analysis for different types of security

Conclusions and Future Work

- Transition to NCW will place a premium on cybersecurity
- Speed of activity in cyberspace calls for automated defenses
- CGFs will have many functions to perform and much remains to be done before they can be fielded
  - Identified requirements and attacks they must be able to manage
- Need to refine requirements and develop distributed CGF system